Advances in Computer Science and Engineering
Volume 1, Number 2, 2007, Pages 119-129
Published online: November 30, 2007
This paper is available online at http://www.pphmj.com

THREE-DIMENSIONAL NUMERICAL COMPUTATION OF FLUID FLOW BY PRECONDITIONING ITERATIVE METHOD

H. HIRANO, H. NIKI[*] and N. OKAMOTO

Department of Applied Chemistry
Okayama Universtiy of Science
1-1, Ridai-cho, Okayama, 700-0005, Japan
e-mail: hirano@dac.ous.ac.jp

[*]Department of Information Science
Okayama Universtiy of Science
1-1, Ridai-cho, Okayama, 700-0005, Japan

Abstract

Three-dimensional numerical computation of natural convection in an enclosure of an incompressible Newtonian fluid was carried out by a preconditioning Gauss-Seidel iterative method using the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm. Preconditioners adopted consist of entries of an original coefficient matrix, and these preconditioners can be obtained without any calculations. The iteration number to solve the linear system of equations of the pressure-correction was reduced to about 42-72% of that without preconditioning. The effect of preconditioning was almost independent of grid number in the present computations.

1. Introduction

The SIMPLE algorithm [6] has been an effective scheme for computational fluid dynamics. This is based on the semi-implicit scheme,

2000 Mathematics Subject Classification: 65H10.

Keywords and phrases: preconditioning iterative method, natural convection, three-dimensional computation.

Received July 18, 2007

and it consists of solving the linear systems of guessed velocity components, pressure-correction and temperature (in non-isothermal problem). Consequently, the most part of computational time is spent on solving linear systems of equations, and the efficiency of solution is crucial. Some variations [1] of the SIMPLE method has been proposed to reduce the total run time from the view point of the computational algorithm. On the other hand, in the present study, the enhancement of computation of fluid flow is investigated on from the view point of solving a linear system of equations. Accordingly, the obtained results could be applicable to other variations of the SIMPLE method.

Paying attention to the solution of a linear system of equations, a set of solution can be obtained with a definite number of operations by the direct solution method (the Gaussian elimination). However, this requires enormous computational memory and time especially in the problem of a large number of grid points such as three-dimensional problem. From this reason, an iterative method is usually adopted. As the acceleration technique of the iterative methods, a positive parameter such as in the SOR (Successive Over-Relaxation) iterative methods and preconditioning can be considered. However, the optimal value of the parameter cannot be estimated before the computation in most of the actual problems such as fluid problems. Accordingly, in the present study, attention is focused on the preconditioning. The preconditioning can be applied to the Krylov subspace methods, and these are essentially different from classical iterative methods in the meaning that the solutions during iterations by use of the Krylov subspace methods are not guaranteed to approach the true solution even if the solution can be finally obtained. On the other hand, transient solutions by classical iterative methods are assured to be asymptotic to the solution. From the above reason, though there are some studies on the application of the Krylov subspace methods [7] at the computation of fluid flow with the SIMPLE algorithm, in the present study, classical iterative methods are adopted and the effect of preconditioning is investigated. Some studies investigated on the application of preconditioning iterative methods to two-dimensional fluid flow [4, 5], and they illustrated that the preconditioners adopted in the present study are effective to reduce the number of iterations. Accordingly, in the present study, the Gauss-Seidel iterative method is adopted, and the effect of preconditioning is studied at calculation of three-dimensional fluid flow problem.

Preconditioning, which can change the degree of diagonal dominance or the condition number of an original linear system of equations, can be considered as an essential strategy to reduce the iteration number. ILU (incomplete LU decomposition) has been widely used as a preconditioner in solving a linear system of equations with Krylov subspace methods [7]. However, certain amount of calculations is required to obtain this type of preconditioner. On the other hand, any calculations do not need to get preconditioners adopted in the present study [2, 4, 5, 8].

The goal of this study is to investigate on the efficiency of the above mentioned preconditioning at the solution of the pressure-correction equation in the SIMPLE algorithm in the three-dimensional fluid flow. Though preconditioners adopted in this study were applied to two-dimensional problem using vorticity equation [4] and the SIMPLE algorithm [5], the present study would be considered valuable in the sense that the degree of diagonal dominance and the spectral radius of a coefficient matrix to solve a three-dimensional problem with the SIMPLE algorithm could be considered different from those of the two-dimensional one.

2. Preconditioning

The following linear system of equations is considered:

$$A\vec{x} = \vec{b}. \tag{1}$$

$A = [a_{i,j}]$ is an $n \times n$ real matrix. Suppose A is an irreducibly diagonally dominant Z-matrix in the present study. Further, A can be split into $A = I - E - F$ without loss of generality. Here, I is the identity matrix, and E and F are the strictly lower and strictly upper parts of $-A$, respectively. $\vec{x}$ and $\vec{b}$ are unknown and given n-dimensional column vectors with components x_i and b_i $(i = 1, 2, \ldots, n)$, respectively. With a preconditioner C, which is also an $n \times n$ real matrix, the preconditioned linear system of equations can be written as follows:

$$CA\vec{x} = C\vec{b}. \tag{2}$$

In the present study, $I + S$ [2] and $I + F$ [8] are adopted as preconditioner C. Here, S consists of the codiagonal entries of $-A$, and can be written as follows:

$$S = \begin{pmatrix} 0 & -a_{1,2} & 0 & \cdots & 0 \\ \vdots & 0 & -a_{2,3} & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \vdots & \vdots & \vdots & 0 & -a_{n-1,n} \\ 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix}. \tag{3}$$

3. Iterative Formulas

Though the Gauss-Seidel iterative method is adopted in the present study, for generality, the iterative formulas of the SOR iterative methods are given as follows:

$$\vec{x}^{(k+1)} = (I - \omega E)^{-1}\{(1-\omega)I + \omega F\}\vec{x}^{(k)} + \omega(I - \omega E)^{-1}\vec{b}, \quad k \geq 0. \tag{4}$$

Here, ω is the relaxation factor. Selecting $\omega = 1$ shows that this iterative method reduces exactly to the Gauss-Seidel iterative method.

Using the preconditioner $C = I + S$, the iterative formula of the preconditioning SOR iterative method can be obtained as follows:

$$\begin{aligned} \vec{x}^{(k+1)} &= \{(I - D_1) - \omega(E + E_1)\}^{-1}\{(1-\omega)(I - D_1) + \omega(F - S + SF)\}\vec{x}^{(k)} \\ &\quad + \omega\{(I - D_1) - \omega(E + E_1)\}^{-1}(I + S)\vec{b}, \quad k \geq 0. \end{aligned} \tag{5}$$

With the preconditioner $C = I + F$, the iterative formula of the preconditioning SOR iterative method can be derived as follows:

$$\begin{aligned} \vec{x}^{(k+1)} &= \{(I - D_2) - \omega(E + E_2)\}^{-1}\{(1-\omega)(I - D_2) + \omega(F_2 + FF)\}\vec{x}^{(k)} \\ &\quad + \omega\{(I - D_2) - \omega(E + E_2)\}^{-1}(I + F)\vec{b}, \quad k \geq 0. \end{aligned} \tag{6}$$

Here, D_1 and D_2 are the diagonal matrices of SE and FE, respectively. E_1 and E_2 are the strictly lower triangular matrices of SE and FE, respectively. F_2 is the strictly upper triangular matrix of FE.

As mentioned above, in the present study, ω was set at unity, and preconditioning Gauss-Seidel iterative methods are used.

4. Numerical Calculation of Fluid Flow

4.1. Governing equations

In the present study, the preconditioning Gauss-Seidel iterative methods described above were applied to the three-dimensional numerical computation of the natural convection in an enclosure. An incompressible Newtonian fluid is assumed, and the governing equations considered are the equations of continuity, motion and energy. These equations can be written as follows in dimensionless form:

$$\nabla \cdot \vec{V} = 0 \text{ in } \Omega, \tag{7}$$

$$\frac{\partial \vec{V}}{\partial T} + (\vec{V} \cdot \nabla)\vec{V} = -\nabla P + \Pr \nabla^2 \vec{V} + \Pr^2 \mathrm{Gr}\Theta(0, 1, 0)^T \text{ in } \Omega, \tag{8}$$

$$\frac{D\Theta}{DT} = \nabla^2 \Theta \text{ in } \Omega. \tag{9}$$

Here, $\Omega \subset \mathbb{R}^3$ is a rectangular domain. Γ is the boundary of a cube shown in Figure 1, and is defined as $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4 \cup \Gamma_5 \cup \Gamma_6$. (x, y, z) are the Cartesian coordinates, and (u, v, w) are the Cartesian components of the velocity vector $\vec{v}$. t, p and θ are the time, pressure and temperature, respectively. A variable in capitals denotes a dimensionless variable of the corresponding lowercase. A dimensional variable with the superscript "*" denotes a reference variable for non-dimensionalization. g is the gravitational acceleration.

Dimensionless variables were defined as follows:

$$X = \frac{x}{x^*},\ Y = \frac{y}{y^*},\ Z = \frac{z}{z^*},\ U = \frac{u}{u^*},\ V = \frac{v}{v^*},\ W = \frac{w}{w^*},$$

$$\Theta = \frac{\theta - \theta^{**}}{\theta^*},\ P = \frac{p}{p^*},\ T = \frac{t}{t^*}.$$

Here, the reference variables can be derived as follows [3]:

$$x^* = y^* = z^* = L,\ u^* = v^* = w^* = \frac{\alpha}{L},\ p^* = \rho\left(\frac{\alpha}{L}\right)^2,\ t^* = \frac{L^2}{\alpha},$$

$$\theta^* = \theta_h - \theta_c,\ \theta^{**} = \frac{\theta_h + \theta_c}{2}.$$

α, β, ν and ρ are the thermal diffusivity, volumetric expansion coefficient, kinematic viscosity and density, respectively. The subscripts of "h" and "c" denote the hot and cold walls as in Figure 1, respectively. Gr and Pr are Grashof and Prandtl numbers, respectively, and are defined as follows:

$$\mathrm{Gr} = \frac{g\beta(\theta_h - \theta_c)L^3}{\nu^2}, \ \mathrm{Pr} = \frac{\nu}{\alpha}. \tag{10}$$

$\nabla^2\vec{V}$ is calculated from the following equation:

$$\nabla^2\vec{V} = \nabla(\nabla \cdot \vec{V}) - [\nabla \times [\nabla \times \vec{V}]].$$

4.2. Computational conditions

Using the dimensionless variables, the computational domain Ω becomes the unit cube:

$$\Omega = \{(X, Y, Z): 0 \le (X, Y, Z) \le 1\}.$$

In the present study, the following uniform cubical grid system is imposed on the unit cube:

$$(X_i, Y_j, Z_k) = \{(ih, jh, kh): h = 1/N, \ \ i, j, k = 0, 1, 2, \ldots, N\}.$$

Here, h is the uniform grid spacing, and the total grid number is equal to N^3. The calculations were carried out for $N = 10, 20, 30, 40$ and 50.

The following conditions are adopted:

$$\begin{cases} U = V = W = 0 & \text{on } \Gamma, \\ \Theta = +0.5 & \text{on } \Gamma_1, \\ \Theta = -0.5 & \text{on } \Gamma_3, \\ \dfrac{\partial\Theta}{\partial Y} = 0 & \text{on } \Gamma_2 \text{ and } \Gamma_4, \\ \dfrac{\partial\Theta}{\partial Z} = 0 & \text{on } \Gamma_5 \text{ and } \Gamma_6, \\ U = V = W = \Theta = 0 \text{ at } T = 0 \text{ and } 0 < (X, Y, Z) < 1, \\ \mathrm{Pr} = 0.71, \mathrm{Gr} = 10^6, \Delta T = 5 \times 10^{-3}. \end{cases}$$

4.3. SIMPLE algorithm

Let the values at n time level be known, and the fields at the next $n + 1$ time level be calculated implicitly by the following equation:

$$\vec{V}^{n+1} = \vec{V}^{n} - \Delta T\{(\vec{V}^{n+1} \cdot \nabla)\vec{V}^{n+1} + \nabla P^{n+1} - \nabla^2 \vec{V}^{n+1}\}. \tag{11}$$

Above governing equations were solved by the SIMPLE algorithm. This algorithm is composed of the following steps:

• Using the guessed pressure field P^*, calculate the guessed velocity $\vec{V}^*$ by the following equation implicitly. The pressure at the former time level is used as P^* in the present study.

$$\vec{V}^{*} = \vec{V}^{n} - \Delta T\{(\vec{V}^{n} \cdot \nabla)\vec{V}^{*} + \nabla P^{*} - \nabla^2 \vec{V}^{*}\}. \tag{12}$$

• Calculate the velocity-correction $\vec{V}' = \vec{V}^{n+1} - \vec{V}^*$ from the following equation by neglecting some terms

$$\vec{V}^{n+1} - \vec{V}^{*} \simeq -\Delta T \nabla (P^{n+1} - P^{*}). \tag{13}$$

• Consequently, the pressure-correction $P' = P^{n+1} - P^*$ can be obtained by the following equation implicitly, since $\nabla \cdot \vec{V}^{n+1} = 0$.

$$\nabla^2 P' = \frac{\nabla \cdot \vec{V}^{*}}{\Delta T}. \tag{14}$$

P' which is necessary to correct the guessed velocity $\vec{V}^*$ from (13) is set at zero on all boundaries, since the non-slip condition is imposed and the velocities should not be corrected on these points in the present computation. However, since the staggered grid system is adopted in the SIMPLE algorithm, the velocity on boundaries is corrected with the normal derivative of P'. Further, the problem of the relativity of pressure should be considered. In the present study, only the Newmann condition for P' was imposed on boundaries, and P' was allowed to seek its own level for faster convergence.

In actual computations, the velocity, pressure and temperature fields are underrelaxed with respect to the previous time level. In the present computation, 0.1, 0.3 and 0.5 are used for this underrelaxation parameters of the pressure-correction, guessed velocity components and temperature, respectively.

5. Result and Discussion

Calculations were carried out for the transient system of Pr = 0.71 and Gr = 10^6.

Table 1 is the comparison of the iteration number to solve the linear system of equations of the pressure-correction. ΔT was set at 0.005, and 200 time steps were advanced, and the calculation was continued till $T = 1$. In the SIMPLE algorithm, the linear systems of equations of the guessed velocity components and temperature are also to be calculated. However, generally, the solution of these equations requires the less number of iterations in comparison with that of the pressure-correction. This is because the linear system of equations of the pressure-correction from (14) is Poisson equation and the coefficient matrix becomes irreducibly weakly diagonally dominant Z-matrix. On the other hand, the coefficient matrix of the guessed velocity components and temperature becomes irreducibly strongly diagonally dominant Z-matrix in the transient problem, since each diagonal entry contains the inverse of time step. The iteration number by the preconditioning Gauss-Seidel iterative method with $I + S$ is reduced to about 66-72% of that without preconditioning. Further, using $I + F$ as preconditioner, the iteration number decreases to about 42-44% of that without preconditioning. The effect of preconditioning seems independent of grid system in the present computations.

Figure 2 shows the transient response of root mean squared velocity components and temperature with the grid system of $N = 50$. Figure 2 illustrates that the flow field becomes almost steady at $T = 1$, and computation was carried out till $T = 1$ in the present study.

6. Conclusion

The effect of preconditioning in the numerical computation of the three-dimensional natural convection in an enclosure by the Gauss-Seidel iterative method with the SIMPLE algorithm is investigated.

The iteration number to solve the linear system of the pressure-correction by the preconditioning Gauss-Seidel iterative method with $I + S$ is reduced to about 66-72% of that without preconditioning.

Using $I+F$ as preconditioner, the iteration number decreases to about 42-44% of that without preconditioning.

The effect of preconditioning is independent of grid system and dimension in the present computations.

References

[1] J. P. Van Doormaal and G. D. Raithby, Enhancements of the SIMPLE method for predicting incompressible fluid flows, Numer. Heat Trans. 7 (1984), 147-163.

[2] A. D. Gunawardena, S. K. Jain and L. Snyder, Modified iterative methods for consistent linear systems, Linear Algebra Appl. 154-156 (1991), 123-143.

[3] J. D. Hellums and S. W. Churchill, Simplification of the mathematical description of boundary and initial value problems, A.I.Ch.E. J. 10 (1964), 110-114.

[4] H. Hirano and H. Niki, Application of a preconditioning iterative method to the computation of fluid flow, Numer. Funct. Anal. Optimiz. 22 (2001), 405-417.

[5] H. Hirano, H. Ozoe, H. Niki and N. Okamoto, An attempt at numerical calculation of natural convection using preconditioned iterative methods, Numer. Heat Trans. A 44 (2003), 97-103.

[6] S. V. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing, Washington, 1980.

[7] S. C. Sheen and J. L. Wu, Solution of the pressure correction equation by the preconditioned conjugate gradient method, Numer. Heat Trans. B 32 (1997), 215-230.

[8] M. Usui, H. Niki and T. Kohno, Adaptive Gauss Seidel method for linear systems, Intern. J. Comput. Math. 51 (1994), 119-125.

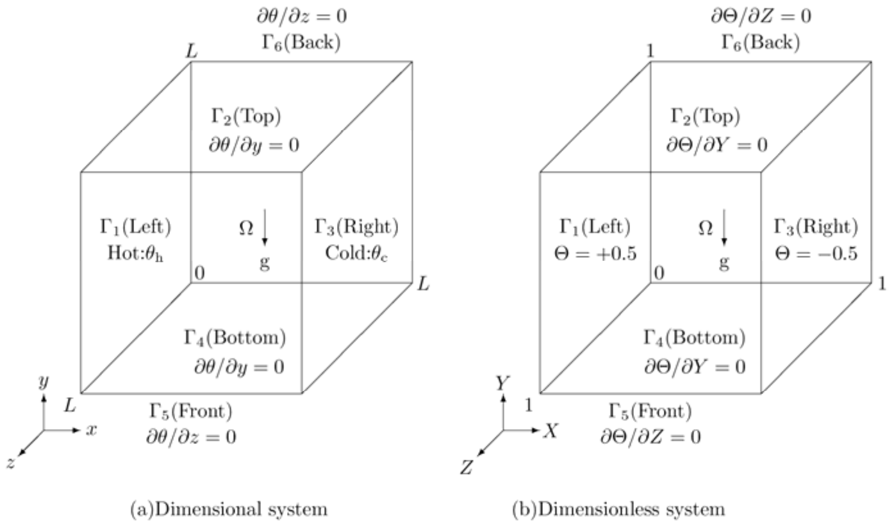

Figure 1. Problem schematic.

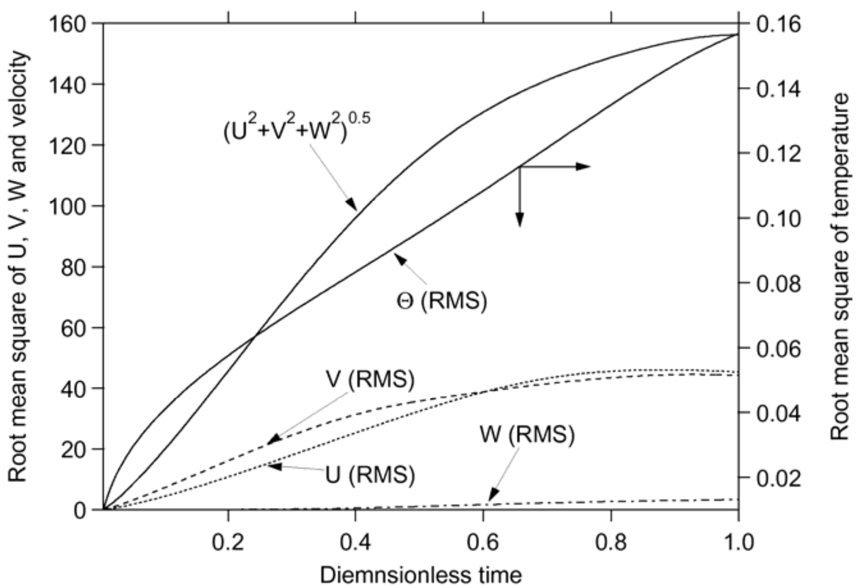

Figure 2. Transient response of root mean square values of the velocity components and temperature at $\mathrm{Pr} = 0.71$ and $\mathrm{Gr} = 10^6$.

Table 1. Comparison of total iteration number to solve the linear system of equations of the pressure-correction with $\Delta T = 0.005$ and 200 time steps for the three-dimensional system at $\mathrm{Pr} = 0.71$ and $\mathrm{Gr} = 10^6$

Grid Number	GS	GS with $I + S$	GS with $I + F$
10^3	32797(1)	21592(0.658)	13955(0.425)
20^3	123189(1)	85866(0.697)	53850(0.437)
30^3	258319(1)	182332(0.706)	114348(0.443)
40^3	392199(1)	273120(0.696)	172925(0.441)
50^3	515383(1)	369065(0.716)	228867(0.444)

The value in brackets is the ratio calculated by dividing the number of iterations by that with the Gauss-Seidel iterative method for each grid system.

Stopping criteria are as follows:

$\|(\vec{b} - A\vec{x})\| / \|\vec{b}\| < 10^{-6}$ for the Gauss-Seidel iterative method, and

$\|(C\vec{b} - CA\vec{x})\| / \|C\vec{b}\| < 10^{-6}$ for the preconditioning Gauss-Seidel iterative method.

■